National Institute of Standards and Technology

Workshop on Building Security Checklists for IT Products
September 25-26, 2003

Panel Session – Deployment and Verification of Checklists

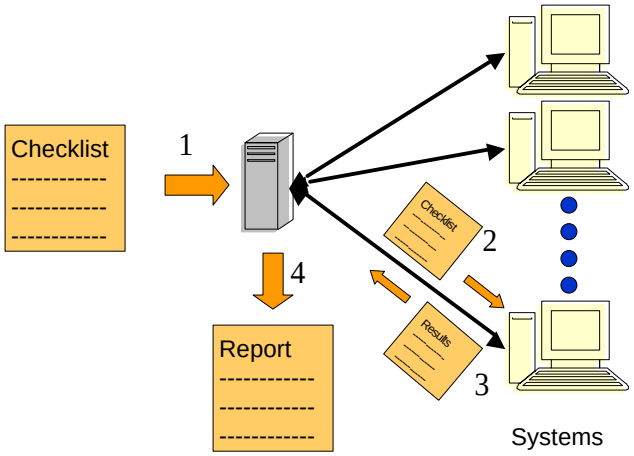
10:40 AM - 11:30 September 26

Dennis R. Moreau
CTO, Configuresoft, Inc.

Deploying and Verifying Checklists

- Panel to present how to use checklists within an environment. The panel will also discuss available tools to assist Security Administrators and Auditors to verify proper application of the checklist.

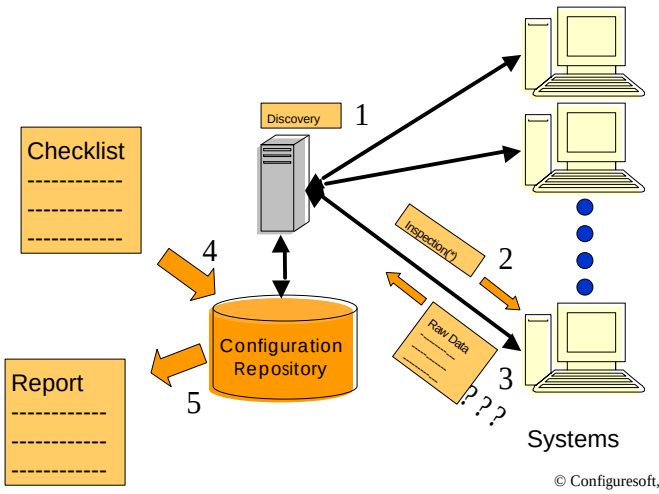
Deployment Architecture (A)



Traditional

- Straight forward for assessment
- Proven in many large scale environments
- May be agent or agent-less
- May include concentrators/brokers/mid-level collectors for scale

Deployment Architecture (B)



A is simpler but …

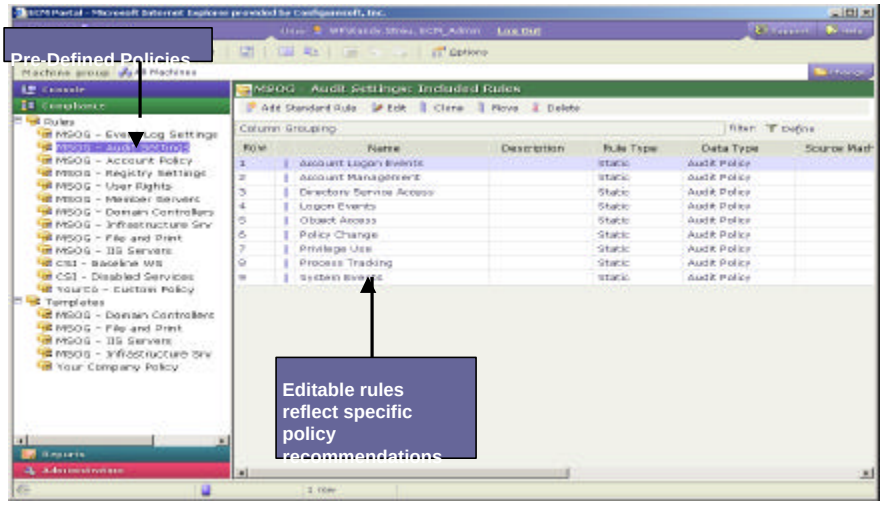
- B closes the loop providing enterprise view of distributed configurations (both
- B supports much shorter assessment times that are independent of distribution times (sampling)
- B provides detailed context and rapid refinement
- B provides for assessment of intersystem configuration consistency
- B provides more rapid detection of configuration "drift" (backup, installs, modification)
- B can provide focused attention to change, independent of checklist formulation
- B provides leveraging of data to IT response (triage & focus) and integration with broad spectrum of IT tools (NSM, HD, TT …)
- B provides for configuration optimization by centralizing characterization and measurement data
- B supports adaptivity: impact assessment, evolution tracking, … of supporting a spectrum of checklists (one size does not fit all)
- B supports change: ancillary parameters, signatures …

IT Configuration Pipe

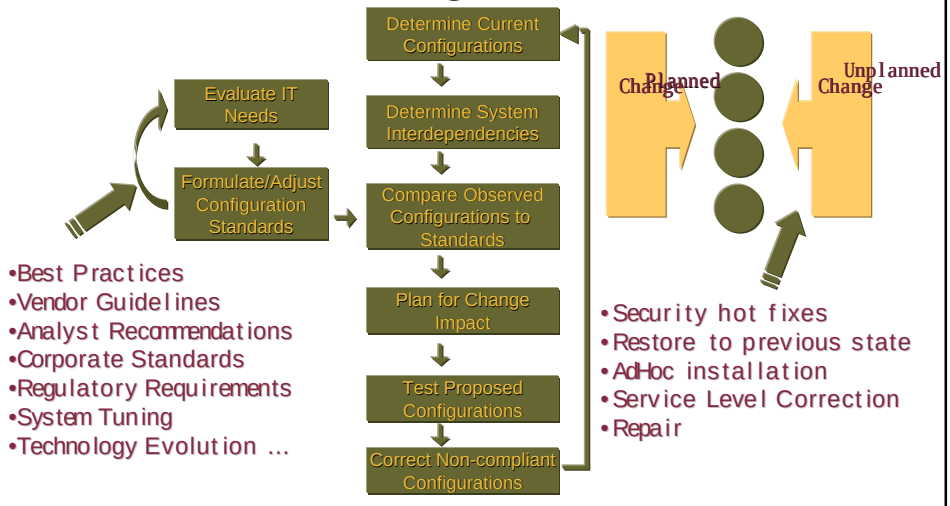
Event Logs
File System
Registry
Accounts
Audit
Group Policy
Processes
Applications
Network
Services
Shares
Hot fixes
ERDs
Groups
…

Discovery → Analysis → Scope, Relevance, Impact, Adaptation → Compliance → OOPS → Mitigation, Remediation, Integration (Event logs, Webish, …)

Security
Performance
Availability
…



The Change Process



Configuration Management is Hard

- Comprehensive configuration information is distributed across thousands of data elements behind scores of APIs
- Different configuration settings are often tightly coupled by application specific requirements
- Configurations may be interdependent
- Configuration changes are often not reversible

Laws of Configuration Management

1. **The actual EC always lags the standard EC**
2. **The standard EC always lags the desired EC**
3. **The desired EC always lags the needed EC**

Scenario 1

- Large energy firm with global IT infrastructure
- 4800 Servers
- SQL Slammer assessment time: 2 min. (discovery, MSDE …)
- Refined mitigation footprint (running service)
- Time to close vulnerability window: 2 hours (see **Remediation Heuristic**)

Remediation Heuristic

- Precisely and rapidly determine vulnerable systems to:
  - **Limit changes to systems "compatible with change"**
  - **Limit footprint of unintended effects**
  - **Close window of vulnerability as quickly as possible**

Scenario 2

- Large telecom firm with evolving IT infrastructure (merger)
- 3000 Servers across 4 data centers
- SNMP buffer overflow assessment time: 3 min.
- Refined mitigation footprint (disable/patch service in exposed segment)
- Time to close vulnerability window: 4 hours

Summary

- Effective checklist assessment is complex and continuous
- Configuration management techniques can improve
  - **assessment times (?T request to report)**
  - **assessment comprehensiveness (inter system)**
  - **remediation focus (reducing window of vulnerability due to evolution/drift)**
  - **remediation efficiency (context)**